1997-05

On bi-decomposition of logic functions

Sasao, Tsutomu

International Workshop on Logic Synthesis, Lake Tahoe, California, May 18-21, 1997, vol.2, Session 8-1, pp.1-6.

http://hdl.handle.net/10945/35844

On Bi-Decompositions of Logic Functions

Tsutomu Sasao

Department of Computer Science
and Electronics
Kyushu Institute of Technology
Iizuka 820, Japan

Jon T. Butler

Department of Electrical and
Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121, U.S.A.

Abstract

A logic function f has a disjoint bi-decomposition iff f can be represented as $f = h(g_1(X_1), g_2(X_2))$, where X_1 and X_2 are disjoint set of variables, and h is an arbitrary two-variable logic fuction. f has a non-disjoint bi-decomposition iff f can be represented as $f(X_1, X_2, x) = h(g_1(X_1, x), g_2(X_2, x))$, where x is the common variable. In this paper, we show a fast method to find bi-decompositions. Also, we enumerate the number of functions having bi-decompositions.

I Introduction

Functional decomposition is a basic technique to realize economical networks. If the function f is represented as $f(X_1, X_2) = h(g(X_1), X_2)$, then f can be realized by the network shown in Fig. 1.1. To find such a decomposition,

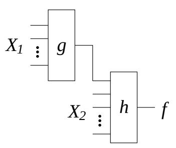

Figure 1.1: A simple disjoint decomposition.

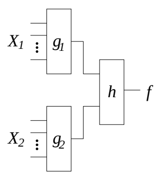

Figure 1.2: A disjoint bi-decomposition.

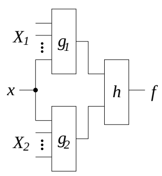

Figure 1.3: A non-disjoint bi-decomposition.

a decomposition chart with 2^{n_1} columns and 2^{n_2} rows are used, where n_i is the number of variables in X_i $(i = 1, 2)$. When n is large, the decomposition chart is too large to build. Recently, a method using BDDs has been developed [13]. This greatly reduces memory requirements and computation time. However, it is still time consuming, since we have to check all the $\binom{n_1+n_2}{n_1}$ partitions of $n = n_1 + n_2$. In this paper, we consider bi-decompositions of logic functions, a restricted class of functional decompositions that have the form $f(X_1, X_2) = h(g_1(X_1), g_2(X_2))$. Fig. 1.2 shows the realization of this decomposition.

The reasons we consider bi-decompositions are as follows:

1) If f has no bi-decomposition, then the computation time is quite small.
2) Some programmable logic devices have two-input logic elements in the outputs.
3) If f has a bi-decomposition, then the optimization of the expression is relatively easy.

A resticted class of bi-decompositions has been considered by [8]. The goals of this paper are

1) Present a fast method for finding bi-decompositions.
2) Enumerate the functions that have bi-decompositions.

Most of the proofs are omitted. They can be available from authors.

II Disjoint Bi-Decomposition

Definition 2.1 *Let $X = (X_1, X_2)$ be a partition of the variables. A logic function f has a disjoint bi-decomposition iff f can be represented as $f(X_1, X_2) = h(g_1(X_1), g_2(X_2))$, where h is any two-variable logic function.*

If f has a disjoint bi-decomposition, then f can be realized by the network shown in Fig. 1.2.

Definition 2.2 *Let $X = (X_1, X_2)$ be a partition of the variables. Let n_1 and n_2 be the number of variables in X_1 and X_2, respectively. A decomposition chart of the function f for a partition (X_1, X_2) consists of 2^{n_1} columns and 2^{n_2} rows of 0s and 1s. The 2^{n_1} distinct binary numbers for X_1 are listed across the top, and the 2^{n_2} distinct binary numbers for X_2 are listed down the side. The entry for the chart corresponds to the value of $f(X_1, X_2)$.*

Example 2.1 *Two decomposition charts for the function $f(x_1, x_2, x_3, x_4) = x_1x_2 \oplus x_3x_4$ are shown in Fig. 2.1 (a) and (b).* □

Note that the decomposition chart is similar to the Karnaugh map with a different ordering for the cell locations.

Definition 2.3 *The number of distinct column (row) patterns in the decomposition chart is called column (row) multiplicity.*

$X_1 = (x_1, x_2)$ (columns), $X_2 = (x_3, x_4)$ (rows)

	00	01	10	11
00	0	0	0	1
01	0	0	0	1
10	0	0	0	1
11	1	1	1	0

(a)

$X_1 = (x_1, x_3)$ (columns), $X_2 = (x_2, x_4)$ (rows)

	00	01	10	11
00	0	0	0	0
01	0	1	0	1
10	0	0	1	1
11	0	1	1	0

(b)

Figure 2.1: Decomposition chart.

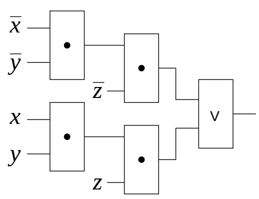

Figure 3.1: A realization of $f(x, y, z) = \bar{x}\bar{y}\bar{z} \vee xyz$.

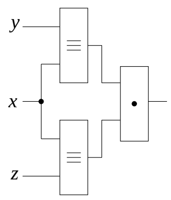

Figure 3.2: Non-disjoint bi-decomposition for $f(x, y, z) = \bar{x}\bar{y}\bar{z} \vee xyz$.

Example 2.2 *In Fig. 2.1 (a), the row and column multiplicities are two. In Fig. 2.1 (b), the row and column multiplicities are four.* □

Definition 2.4 *Let $\mu(f : X_1, X_2)$ be the column multiplicities for f with respect to X_1 and X_2. Let $\mu(f : X_2, X_1)$ be the row multiplicities for f with respect to X_1 and X_2.*

Theorem 2.1 *f has a disjoint bi-decomposition of form $f(X_1, X_2) = h(g_1(X_1), g_2(X_2))$ iff $\mu(f : X_1, X_2) \leq 2$ and $\mu(f : X_2, X_1) \leq 2$.*

III Non-Disjoint Bi-Decomposition

Definition 3.1 *Let X_1 and X_2 be disjoint sets of variables, and let x be disjoint from X_1 and X_2. A logic function f has a non-disjoint bi-decomposition iff f can be represented as $f(X_1, X_2, x) = h(g_1(X_1, x), g_2(X_2, x))$, where h is a two-variable logic function. In this case, x is called the common variable.*

A function f with a non-disjoint bi-decomposition can be realized by the network shown in Fig. 1.3.

Lemma 3.1 *Let $X = (X_1, X_2, x)$ be a partition of the input variables. Let $h(g_1, g_2)$ be an arbitrary logic function of two variables. Then,*

$$h(g_1(X_1, x), g_2(X_2, x)) = \bar{x}h(g_1(X_1, 0), g_2(X_2, 0)) \vee xh(g_1(X_1, 1), g_2(X_2, 1)).$$

Definition 3.2 *Let x be the common variable of the non-disjoint bi-decomposition. Let $f(X_1, X_2, a)$ be a sub-function, where x is set to a 0 or 1.*

$X_1 = (x_1, x_2)$ (columns), $X_2 = (x_3, x_4)$ (rows)

	00	01	10	11
00	1	0	0	0
01	1	0	0	0
10	1	0	0	0
11	0	1	1	1

(a) $f_0 = \bar{x}_1\bar{x}_2 \oplus x_3x_4$

$X_1 = (x_1, x_2)$ (columns), $X_2 = (x_3, x_4)$ (rows)

	00	01	10	11
00	0	0	0	1
01	1	1	1	0
10	1	1	1	0
11	1	1	1	0

(b) $f_1 = x_1x_2 \oplus (x_3 \vee x_4)$

Figure 3.3: Functions in Example 3.2

Theorem 3.1 *$f(X_1, X_2, x)$ has a non-disjoint bi-decomposition of the form $h(g_1(X_1, x), g_2(X_2, x))$ iff $f(X_1, X_2, 0)$ and $f(X_1, X_2, 1)$ have disjoint bi-decompositions $h(g_{01}(X_1), g_{02}(X_2))$ and $h(g_{11}(X_1), g_{12}(X_2))$, respectively.*

Example 3.1 *Consider the three-variable function: $f(x, y, z) = \bar{x}\bar{y}\bar{z} \vee xyz$. Suppose modules that realizes any function of two variables can be used. The straightforward realization shown in Fig. 3.1 requires five modules. The Shannon expansion with respect to x is $f(x, y, z) = \bar{x}f(0, y, z) \vee xf(1, y, z)$, where $f(0, y, z) = \bar{y}\bar{z}$, and $f(1, y, z) = yz$. Note that both $f(0, y, z)$ and $f(1, y, z)$ have bi-decompositions with $h(x, y) = xy$. Since, $g_1(x, y) = \bar{x}g_{01}(X_1) \vee xg_{11}(X_1) = \bar{x}\bar{y} \vee xy$, and $g_2(x, y) = \bar{x}g_{02}(X_2) \vee xg_{12}(X_2) = \bar{x}\bar{z} \vee xz$. We have $f(x, y, z) = g_1(x, y)g_2(x, z) = (\bar{x}\bar{y} \vee xy)(\bar{x}\bar{z} \vee xz)$. From this expression, we have the network in Fig. 3.2. This network requires only three modules.* □

Example 3.2 *Consider the five-variable function $f = \bar{x}_5f_0 \vee x_5f_1$, where f_0 and f_1 are shown in Fig. 3.3. Since both f_0 and f_1 have disjoint bi-decompositions of the form $h(g_1(X_1), g_2(X_2))$, $f = \bar{x}_5f_0 \vee x_5f_1$ has a non-disjoint bi-decomposition as follows:*

$$\begin{aligned} f &= \bar{x}_5\{\bar{x}_1\bar{x}_2 \oplus x_3x_4\} \vee x_5\{x_1x_2 \oplus (x_3 \vee x_4)\} \\ &= \{\bar{x}_5(\bar{x}_1\bar{x}_2) \vee x_5(x_1x_2)\} \oplus \{\bar{x}_5(x_3x_4) \vee x_5(x_3 \vee x_4)\}. \end{aligned}$$

The converse is true also. □

Up to now, we only considered the case where there is a single common variable. However, the theorem can be extened to k common variables, where $k \geq 2$.

Definition 3.3 *Let X_1, X_2, and X_3 be disjoint sets of variables. Let $f(X_1, X_2, \boldsymbol{a})$ be the sub-functions, where X_3 is set to $\boldsymbol{a} \in \{0, 1\}^k$, and k denotes the number of variables in X_3.*

Theorem 3.2 *Let X_1, X_2, and X_3 be disjoint sets of variables. Then, f has a non-disjoint bi-decomposition of form $f(X_1, X_2, X_3) = h(g_1(X_1, X_3), g_2(X_2, X_3))$ iff $f(X_1, X_2, \boldsymbol{a})$ has a decomposition of the form $h(g_{1\boldsymbol{a}}(X_1), g_{2\boldsymbol{a}}(X_2))$ for all possible $\boldsymbol{a} \in \{0, 1\}^k$, where k denotes the number of variables in X_3.*

IV A Fast Method for Bi-Decompositions

In this section, we show necessary and sufficient conditions for a function to have a disjoint bi-decomposition. Then, we show efficient algorithms to find disjoint bi-decompositions. In the previous sections, $h(g_1, g_2)$ is an arbitrary two-variable logic function. To find a disjoint bi-decomposition, we need to consider only three types:

1) OR type: $f = g_1(X_1) \vee g_2(X_2)$,
2) AND type: $f = g_1(X_1)g_2(X_2)$, and
3) EXOR type: $f = g_1(X_1) \oplus g_2(X_2)$.

Since f has an AND type disjoint bi-decomposition iff $\bar{f}$ has OR type disjoint bi-decomposition, we only consider the OR type and EXOR type bi-decompositions.

Definition 4.1 *x and $\bar{x}$ are literals of a variable x. A logical product which contains at most one literal for each variable is called a product term or a product. Product terms combined with OR operators form a sum-of-products expression (SOP).*

Definition 4.2 *A prime implicant (PI) p of a function f is a product term which implies f, such that the deletion of any literal from p results in a new product which does not imply f.*

Definition 4.3 *An irredundant sum-of-products expression (ISOP) is an SOP, where each product is a PI, and no product can be deleted without changing the function represented by the expression.*

Definition 4.4 *Let $f(X)$ be a function and p be a product of literal(s) in X. The restriction of f to p, denoted by $f(X|p)$ is obtained as follows: If x_i appears in p, then set x_i in 1 in f, and if $\bar{x}_i$ appears in p, then set x_i in 0 in f.*

Example 4.1 *Let $f(x_1, x_2, x_3) = x_1x_2 \vee \bar{x}_2x_3$ and $p = x_1x_3$. $f(X|p)$ is obtained as follows: Set $x_1 = x_3 = 1$ in f, and we have $f(X|x_1x_3) = f(1, x_2, 1) = x_2 \vee \bar{x}_2 = 1$.* □

Lemma 4.1 *p is an implicant of $f(X)$, iff $f(X|p) = 1$.*

Example 4.2 *By Lemma 4.1, x_1x_3 is an implicant of $x_1x_2 \vee \bar{x}_2x_3$, shown in Example 4.1.* □

Theorem 4.1 *(OR type disjoint bi-decomposition) f has a disjoint bi-decomposition of form $f(X_1, X_2) = g_1(X_1) \vee g_2(X_2)$ iff every product in an ISOP for f consists of literals from X_1 only or X_2 only.*

Definition 4.5 $x^0 = \bar{x}$. $x^1 = x$.

Corollary 4.1 *If $f(x_1, x_2, \ldots, x_n)$ has a PI of the form $x_1^{a_1}x_2^{a_2}\cdots x_n^{a_n}$, where $a_i \in \{0,1\}$, then f has no OR type disjoint bi-decomposition.*

Let $x_i (i = 1, 2, \ldots, n)$ be the input variables of f. Let $p_1 \vee p_2 \vee \cdots \vee p_t$ be an irredundant sum-of-products expression for f, where p_i $(i = 1, 2, \ldots, t)$ are PIs of f. Let Π_0 be the trivial partition of $\{1, 2, \ldots, n\}$, $\Pi_0 = [\{1\}, \{2\}, \ldots, \{n\}]$.

Algorithm 4.1 *(OR type disjoint bi-decomposition: $f(X_1, X_2) = g_1(X_1) \vee g_2(X_2)$).*

1. *For $i = 1$ to t, form Π_i from Π_{i-1} by merging two blocks Ω_1 and Ω_2 of Π_{i-1} if at least one literal in p_i occurs in both Ω_1 and Ω_2.*
2. *If Π_t has at least two blocks, then $f(X_1, X_2)$ has a disjoint bi-decomposition of the form $f(X_1, X_2) = g_1(X_1) \vee g_2(X_2)$, with X_1 the union of one or more blocks of Π_t and X_2 the union of the remaining blocks.*

Example 4.3 *Consider the ISOP: $f(x_1, x_2, \ldots, x_6) = x_1x_2 \vee x_2x_3 \vee x_4x_5 \vee x_5x_6$. The products x_1x_2 and x_2x_3 show that x_1, x_2, and x_3 are in the same block. Also, the products x_4x_5 and x_5x_6 show that x_4, x_5, and x_6 are in the same block. Thus, we have the partition $[\{1, 2, 3\}, \{4, 5, 6\}]$. The corresponding OR type disjoint bi-decomposition is $f(X_1, X_2) = g_1(X_1) \vee g_2(X_2)$, where $X_1 = (x_1, x_2, x_3)$ and $X_2 = (x_4, x_5, x_6)$.* □

Example 4.4 *Consider the function f with an ISOP: $f(x_1, x_2, x_3, x_4, x_5) = x_1x_2x_3 \vee x_3x_4x_5$.*

1) *The product $x_1x_2x_3$ shows that x_1, x_2, and x_3 belong to the same block.*
2) *The product $x_3x_4x_5$ shows that x_3, x_4, and x_5 belong to the same block.*

Thus, all the variables belong to the same block. From this, it follows that f has no OR type decomposition. □

Theorem 4.2 *(AND type disjoint bi-decomposition) f has a disjoint bi-decomposition of form $f(X_1, X_2) = g_1(X_1)g_2(X_2)$ iff every product in an ISOP for $\bar{f}$ consists of literals from X_1 only or X_2 only.*

Lemma 4.2 *[15] An arbitrary n-variable function can be uniquely represented as*

$$f(x_1, x_2, \ldots, x_n) = a_0 \oplus (a_1x_1 \oplus a_2x_2 \oplus \cdots \oplus a_nx_n) \oplus(a_{12}x_1x_2 \oplus a_{13}x_1x_3 \oplus \cdots \oplus a_{n-1n}x_{n-1}x_n) \oplus \cdots \oplus a_{12\cdots n}x_1x_2\cdots x_n, \quad (4.1)$$

where $a_i \in \{0,1\}$. The above expression is called a positive polarity Reed-Muller expression (PPRM).

For a given function f, the coefficients $a_0, a_1, a_2, \ldots, a_{12\cdots n}$ are uniquely determined. Thus, the PPRM is a canonical representation. The number of products in (4.1) is at most 2^n, and all the literals are positive (uncomplemented).

Theorem 4.3 *(EXOR type disjoint bi-decomposition) f has a disjoint bi-decomposition of the form $f(X_1, X_2) = g_1(X_1) \oplus g_2(X_2)$ iff every product in the PPRM for f consists of literals from X_1 only or X_2 only.*

Corollary 4.2 *If the PPRM of an n-variable function has the product $x_1 x_2 \cdots x_n$, then f has no EXOR type disjoint bi-decomposition.*

Theorem 4.4 *When f has an EXOR type disjoint bi-decomposition, the number of true minterms of f is an even number.*

Corollary 4.3 *When the number of true minterms of f is an odd number, then f does not have an EXOR type disjoint bi-decomposition.*

The significance of this observation is that at least one half of the functions can be quickly rejected as candidates for EXOR type disjoint bi-decomposition.

Let x_i $(i = 1, 2, \ldots, n)$ be the input variables of f. Let $p_1 \oplus p_2 \oplus \cdots \oplus p_t$ be PPRM for f, where p_i $(i = 1, 2, \ldots, t)$ are products. Let, Π_0 be the trivial partition of $\{1, 2, \ldots, n\}$, $\Pi_0 = [\{1\}, \{2\}, \ldots, \{n\}]$.

Algorithm 4.2 *(EXOR type disjoint bi-decomposition: $f(X_1, X_2) = g_1(X_1) \oplus g_2(X_2)$).*

1. *For $i = 1$ to t, form Π_i from Π_{i-1} by merging two blocks Ω_1 and Ω_2 of Π_{i-1} if at least one literal in p_i occurs in both Ω_1 and Ω_2.*
2. *If Π_t has at least two blocks, then $f(X_1, X_2)$ has a disjoint bi-decomposition of form $f(X_1, X_2) = g_1(X_1) \oplus g_2(X_2)$, with X_1 the union of one or more blocks of Π_t and X_2 the union of the remaining blocks.*

Example 4.5 *Consider the PPRM: $f(x_1, x_2, \ldots, x_6) = x_1x_2 \oplus x_2x_3 \oplus x_4x_5 \oplus x_5x_6$. The products x_1x_2 and x_2x_3 show that x_1, x_2, and x_3 are in the same block. Also, the products x_4x_5 and x_5x_6 show that x_4, x_5, and x_6 are in the same block. Thus, we have the partition $[\{1, 2, 3\}, \{4, 5, 6\}]$. The corresponding EXOR type disjoint bi-decomposition is $f(X_1, X_2) = g_1(X_1) \oplus g_2(X_2)$, where $X_1 = (x_1, x_2, x_3)$ and $X_2 = (x_4, x_5, x_6)$.* □

Algorithm 4.3 *(Non-disjoint bi-decomposition). $f(X_1, X_2, x_i) = g_1(X_1, x_i) \otimes g_2(X_2, x_i)$, where $\otimes$ denotes either OR, AND, or EXOR. Let (X_1, X_2, x_i) be a partition of the variables x_1, x_2,..., and x_n. For $i = 1$ to n, do*

i) Let $f_{0i} = f(X_1, X_2, 0)$. (Set x_i to 0). Let $f_{1i} = f(X_1, X_2, 1)$. (Set x_i to 1).

ii) If both f_{0i} and f_{1i} have the same type of disjoint bi-decompositions with the same partition, then f has a non-disjoint bi-decomposition.

V Complexity Analysis of the Algorithms

5.1 OR type disjoint bi-decomposition

We assume that the function is given as an ISOP with t products. Note that $t \leq 2^{n-1}$. The time to form the partition of variables is $O(n \cdot t)$.

5.2 EXOR type disjoint bi-decomposition

A PPRM can be represented by a functional decision diagram (FDD [5, 15]). Each path from the root node to the constant 1 node corresponds to a product in the PPRM. Thus, the partition of the input variables is directly generated from the FDD. The number of paths in an FDD is $O(2^n)$, where n is the number of the input variables. However, we can avoid exhaustive generation of paths as follows: Let p_1 and p_2 be products in a PPRM. If all the literals in p_1 also appear in p_2, then p_2 need not be generated in the Algorithm, since the product p_1 that contains more literals than p_2 is more important. By searching the paths with more literals first, we can efficiently detect functions with no disjoint bi-decomposition.

Example 5.1 *Consider the function $f(X)$ given as a PPRM: $f(X) = x_1 \oplus x_1x_2 \oplus x_3x_4 \oplus x_1x_2x_5x_6$. In constructing the partition of X, we need not consider the products x_1 or x_1x_2, since $x_1x_2x_5x_6$ has the literals of x_1 and x_1x_2. In this case, the product $x_1x_2x_5x_6$ shows that x_1, x_2, x_5, and x_6 belong to the same group. Also, the product x_3x_4 shows that x_3 and x_4 belong to the same group. Thus, X is partitioned as $X = (X_1, X_2)$, where $X_1 = (x_1, x_2, x_5, x_6)$ and $X_2 = (x_3, x_4)$.* □

Definition 5.1 *Let p be a product. The set of variables in p is denoted by $V(p) = \{x_i | x_i$ or $\bar{x}_i$ appears in $p\}$. For example, $V(x_1x_2\bar{x}_4) = \{x_1, x_2, x_4\}$*

Definition 5.2 *Let F be a PPRM. A product p is said to have maximal variable set $V(p)$ if there is no other product p' such that $V(p) \subset V(p')$.*

Example 5.2 *For the PPRM, $F = x_1x_2 \oplus x_1x_3 \oplus x_1x_2x_3 \oplus x_4$, $V(x_1x_2) = \{x_1, x_2\}$, $V(x_1x_3) = \{x_1, x_3\}$, $V(x_1x_2x_3) = \{x_1, x_2, x_3\}$, and $V(x_4) = \{x_4\}$. Thus, $x_1x_2x_3$ and x_4 have maximal variable sets.* □

Theorem 5.1 *A function f has an EXOR type disjoint bi-decomposition if a function f' from the PPRM of f by eliminating implicants not having maximal variable sets has an EXOR type disjoint bi-decomposition.*

The following theorem says that if a function has an EXOR type disjoint bi-decomposition, then the number of products in the PPRM is relatively small.

Theorem 5.2 *If f has a disjoint bi-decomposition of the form $f(X_1, X_2) = g_1(X_1) \oplus g_2(X_2)$, then the number of products in the PPRM is at most $2^{n_1} + 2^{n_2} - 1$, where n_i is the number of variables in X_i $(i = 1, 2)$.*

VI Number of Functions with Bi-Decompositions

6.1 Functions with a small number of variables

In the previous sections, we showed that disjoint bi-decompositions are easy to find. In this section, we will enumerate the functions with disjoint bi-decompositions.

Definition 6.1 *A function f is said to be nondegenerate if for all x_i $f(|\bar{x}_i) \neq f(|x_i)$.*

Definition 6.2 *Two functions f and g are NP-equivalent, denoted by $f \overset{\mathrm{NP}}{\sim} g$, iff g is derived from f by the following operations:*

1) Permutation of the input variables.

2) Negations of the input variables.

The following is easy to prove.

Lemma 6.1 *If f has a disjoint bi-decomposition and if $f \overset{\mathrm{NP}}{\sim} g$, then g has also the same type of disjoint bi-decomposition.*

Lemma 6.2 *All the two-variable functions have disjoint bi-decompositions.*

Example 6.1 *There are $2^{2^3} = 256$ three-variable logic functions of which 218 are nondegenerate. These nondegenerate functions are grouped into 16 NP-equivalence classes as shown in Table 6.1 [9]. In this table, the column headed by N denotes the number of functions in that equivalence class. Eight classes have disjoint bi-decompositions, and three have non-disjoint bi-decompositions. Note that 194 functions have bi-decompositions.* □

The number of functions with AND type disjoint bi-decompositions is equal to the number of functions with OR type disjoint bi-decompositions.

In the case of disjoint bi-decompositions, a function has exactly one type of decomposition (Lemma 6.4). On the other hand, in the case of non-disjoint bi-decompositions, a function may have more than one type of bi-decompositions.

Example 6.2 *Consider the three-variable function $f = \bar{x}_1x_3 \vee x_1x_2$. This function has three types of non-disjoint bi-decompositions:*

$f = \bar{x}_1x_3 \vee x_1x_2$ (OR type bi-decomposition)

$= \bar{x}_1x_3 \oplus x_1x_2$ (EXOR type bi-decomposition)

$= (x_1 \vee x_3)(\bar{x}_1 \vee x_2)$ (AND type bi-decomposition) □

Table 6.1: NP-representative functions of three variables.

	Representative functions	N	Type	Property
1	$x_1 \oplus x_2 \oplus x_3$	2	EXOR	Disjoint Bi-Decomposition
2	$x_1x_2x_3$	8	AND	
3	$x_1 \vee x_2 \vee x_3$	8	OR	
4	$x_1(x_2 \vee x_3)$	24	AND	
5	$x_1 \vee x_2x_3$	24	OR	
6	$x_1(x_2 \oplus x_3)$	12	AND	
7	$x_1 \vee (x_2 \oplus x_3)$	12	OR	
8	$x_1 \oplus x_2x_3$	24	EXOR	
9	$x_1x_2x_3 \vee \bar{x}_1\bar{x}_2\bar{x}_3$	4		Non-Disjoint Bi-Decomposition
10	$(x_1 \vee x_2 \vee x_3)(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$	4		
11	$\bar{x}_1x_3 \vee x_1x_2$	24		
12	$x_1\bar{x}_2\bar{x}_3 \vee x_2x_3$	24		
13	$(x_1 \vee \bar{x}_2 \vee \bar{x}_3)(x_2 \vee x_3)$	24		
14	$x_1x_2 \vee x_2x_3 \vee x_3x_1$	8		No Bi-Decomposition
15	$x_1x_2 \vee x_2x_3 \vee x_1x_3 \vee \bar{x}_1\bar{x}_2\bar{x}_3$	8		
16	$\bar{x}_1x_2x_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2\bar{x}_3$	8		

N: Number of the functions in the class.

Table 6.2: Number of functions.

			$n=2$	$n=3$	$n=4$
All the functions			16	256	65536
Nondegenerate functions			10	218	64594
Functions with bi-decomposition	Disjoint	AND	4	44	1660
		OR	4	44	1660
		EXOR	2	26	914
	Non-disjont		0	80	3860
	Total		10	194	8094

6.2 The number of functions with bi-decompositions

Lemma 6.3 *[4]: Let $\alpha(n)$ be the number of nondegenerate functions on n variables. Then,*

$$\alpha(n) = \sum_{k=0}^{n} \binom{n}{k} (-1)^{n-k} 2^{2^k} \sim 2^{2^n},$$

where $a(n) \sim b(n)$ means $\lim_{n\to\infty} \frac{a(n)}{b(n)} = 1$.

Lemma 6.4 *A nondegenerate function f has at most one type of disjoint bi-decomposition:*

1. *$f(X_1, X_2) = g_1(X_1) \cdot g_2(X_2)$,*
2. *$f(X_1, X_2) = g_1(X_1) \vee g_2(X_2)$, or*
3. *$f(X_1, X_2) = g_1(X_1) \oplus g_2(X_2)$,*

where g_1 and g_2 are nondegenerate functions on one or more variables.

Theorem 6.1 *The number of functions $N_{disjoint}(n)$ with disjoint bi-decompositions is $N_{disjoint}(n) = A_{dis}(n) + O_{dis}(n) + E_{dis}(n)$, where*

$$A_{dis}(n) = n! \sum_{\substack{k_1,k_2,\ldots,k_n \geq 0 \\ 1k_1+2k_2+\cdots+nk_n = n}} \prod_{i=1}^{n} \left(\frac{\alpha(i) - A_{dis}(i)}{i!}\right)^{k_i} \frac{1}{k_i!}$$

Table 7.1: Number of functions with bi-decompositions.

Decomposition Type		Number of Functions
Disjoint	AND	853
	OR	264
	EXOR	73
Non-disjoint	AND	162
	OR	91
	EXOR	42

$$O_{dis}(n)=n! \sum_{\substack{k_1,k_2,\ldots,k_n \geq 0 \\ 1k_1+2k_2+\cdots+nk_n=n}} \prod_{i=1}^{n} \left(\frac{\alpha(i)-O_{dis}(i)}{i!}\right)^{k_i} \frac{1}{k_i!}$$

$$E_{dis}(n)=2n! \sum_{\substack{k_1,k_2,\ldots,k_n \geq 0 \\ 1k_1+2k_2+\cdots+nk_n=n}} \prod_{i=1}^{n} \left(\frac{\alpha(i)-E_{dis}(i)}{i!}\right)^{k_i} \frac{1}{2^{k_i}k_i!}$$

where the sums are over all partitions of n except the trivial partition $n = 0\cdot 1 + 0\cdot 2 + \cdots + 0\cdot(n-1) + 1\cdot n$ (i.e. the sum is over all partitions where $k_n = 0$), and where $A_{dis}(1) = O_{dis}(1) = E_{dis}(1) = 0$.

Table 6.2 shows the number of functions with disjoint bi-decompositions up to $n = 4$.

VII Experimental Results

We analyzed the bi-decomposability of 136 benchmark functions. Over these multiple-output functions, the total number of outputs (functions) is 1908. For each function, we determined whether there exists a disjoint bi-decomposition. If none existed, we determined if there exists a non-disjoint bi-decomposition (with a single common variable). Table 7.1 summarizes our results. It is interesting that 1190 out of 1908 functions, or 62 percent, have disjoint bi-decompositions. Of the remaining 718 functions, 295 have non-disjoint decompositions. It should be noted that more than 295 functions have non-disjoint decompositions, since a function with a disjoint bi-decomposition may also have a non-disjoint bi-decomposition.

VIII Conclusions and Comments

In this paper, we presented the bi-decomposition, a special case of functional decomposition. Disjoint bi-decompositions have the following features:

1) They are easy to detect; we use ISOPs or PPRMs rather than decomposition charts.
2) Programmable logic devices exist that realize bi-decompositions.
3) If the function has an OR (AND) type bi-decomposition, then we can optimize the expression separately.

We enumerated functions with bi-decompositions. Among 218 nondegenerate functions of 4 variables, 194 have bi-decompositions. Also, we derived formulae for the number of disjoint bi-decompositions.

Since the fraction of functions with decompositions approaches to zero as n increase [4], the fraction of functions with bi-decompositions also approaches to zero as n increases. However, for 1908 functions we analyzed about 78% of them had either disjoint or non-disjoint bi-decompositions.

References

[1] R. L. Ashenhurst, "The decomposition of switching functions," *In Proceedings of an International Symposium on the Theory of Switching*, pp. 74-116, April 1957.

[2] J. T. Butler, "On the number of functions realized by cascades and disjunctive networks," *IEEE Trans. Comput.*, Vol. C-24, pp. 681-690, July 1975.

[3] H. A. Curtis, *A New Approach to the Design of Switching Circuits*, Princeton, N.J.: Van Nostrand, 1962.

[4] M. A. Harrison, *Introduction to Switching and Automata Theory*, McGraw-Hill, 1965.

[5] U. Kebschull, E. Schubert and W. Rosenstiel, "Multilevel logic synthesis based on functional decision diagrams," *EDAC 92*, 1992, pp. 43-47.

[6] Y-T. Lai, M. Pedram, S. B. K. Vrudhula, "EVBDD-based algorithm for integer linear programming, spectral transformation, and functional decomposition," *IEEE Trans. CAD*, Vol. 13, No. 8. Aug. 1994, pp. 959-975.

[7] A. A. Malik, D. Harrison, and R. K. Brayton, "Three-level decomposition with applicatioin to PLDs," *ICCD-1991*, pp. 628-633, Oct. 1991.

[8] Y. Matsunaga, "An attempt to factor logic functions using exclusive-or decomposition," *SASIMI'96*, pp. 78-83.

[9] S. Muroga, *Logic Design and Switching Theory*, John Wiley & Sons, 1979.

[10] T. Sasao and K. Kinoshita, "On the number of fanout-free functions and unate cascade functions," *IEEE Trans. on Comput.*, Vol. C-28, No. 1, pp. 866-72, Jan. 1979.

[11] T. Sasao, "Input variable assignment and output phase optimization of PLA's," *IEEE Trans. Comput.*, Vol. C-33, No. 10, pp. 879-894, Oct. 1984.

[12] T. Sasao, "Application of multiple-valued logic to a serial decomposition of PLA's," *International Symposium on Multiple-Valued Logic*, Guangzhou, China, pp. 264-271, May 1989.

[13] T. Sasao (ed.), *Logic Synthesis and Optimization*, Kluwer Academic Publishers, 1993.

[14] T. Sasao, "A design method for AND-OR-EXOR three-level networks," *ACM/IEEE International Workshop on Logic Synthesis*, Tahoe City, California, May 23-26, 1995, pp.8:11- 8:20.

[15] T. Sasao and M. Fujita (ed.), *Representations of Discrete Functions*, Kluwer Academic Publishers, 1996.

[16] T. Sasao and J. T. Butler, "Comparison of the worst and best sum-of-products expressions for multiple-valued functions," (preprint).

[17] M. Sauerhoff, I. Wegener, and R. Werchner, "Optimal ordered binary decision diagrams for tree-like circuits," *SASIMI'96* (to be published).

[18] S. Yang, "Logic synthesis and optimization benchmark user guide, version 3.0," *MCNC*, Jan. 1991.